

tion table. Finally, the object parser will update the object-fragment table and fragment-object table accordingly. For each fragment deleted from the fragment description table, the fragment cache manager will be invoked to check if any of its fragment version is in the fragment cache and delete it.

FIG. 13 depicts an example of the client logic. In step 1305, the client waits for input (request from a user or a response from the server). Depending upon the type of input, the appropriate routine will be invoked. If in step 1310, the input is an object request from the user, the request is sent to the server in step 1315 (see FIG. 6) where persistent object fragments in the object are identified and the object revised as necessary.

In step 1320, if the input is an object (e.g., a server response from a previous object request), the object is rendered and displayed to the user in step 1330. Recall that since persistent object fragments have been recognized to make the revised object document cacheable at the server or client device and/or processable/interpretable at the client device. Consider the example where a segment can not be rendered from the markup language description by a simple client device such as WINDOWS CE™-based Internet appliances. According to the present invention, by recognizing the segment as a separate object fragment, the client can process and/or cache the revised document and allow the server to interpret the markup language describing the fragment and generate an appropriate version for the client. Examples of the limitations on the client device include but are not limited to the processing power or storage capacity of the client device to interpret the markup language and generate the object fragment; and/or the bandwidth available to the client device to retrieve the description of the fragment. Recall also that the recognition and revision of an object to remove segments qualifying as object fragments enable the object fragment to be cached separately and reused to avoid going through interpreting the markup language description of the object to generate it again. This will improve response time and reduce server load on fragment re-references. Each fragment—once removed—may need to be requested separately with additional requests from the client. Thus, preferably, only a segment or group of segments that meet a certain threshold on the processing requirements of interpreting the markup language description to generate the object segment were recognized as a fragment by the server.

In step 1335, the client determines whether the object is cacheable. Recall that any dynamic object or object exceeding a certain size will be deemed not cacheable at the client device, which often has limited caching capacity. According to the present invention, the server uses persistent object fragment identifiers to replace persistent object fragments (such as dynamic objects or large segments) in a Web object. The revised object is thus more cacheable at the client device, since the server has removed the dynamic or large objects from the object and reduced the size of the object. For example, recall the example of an object description for a purchase order that includes a dynamic part for retrieving the current price of a product from the database. This dynamic part may be a small portion of the purchase order, but would prevent the object from being cached. According to one feature of the present invention for recognizing and treating the dynamic part as a separate fragment from the object description, the revised document becomes static and therefore cacheable. In step 1340, if the object is cacheable, the object is cached at the local client cache. In step 1325, a miscellaneous routine is invoked to handle other types of input, such as a pager message.

A preferred embodiment of the present invention includes features implemented as software tangibly embodied on a computer program product or program storage device for execution on a processor (not shown) provided with the client (60 . . . 63) and/or server (30 . . . 33). For example, software implemented in a popular object-oriented computer executable code such as JAVA provides portability across different platforms. Those skilled in the art will appreciate that other procedure-oriented and object-oriented (OO) programming environments, including but not limited to C++ and Smalltalk can also be employed.

Those skilled in the art will also appreciate that methods of the present invention may be implemented as software for execution on a computer or other processor-based device. The software may be embodied on a magnetic, electrical, optical, or other persistent program and/or data storage device, including but not limited to: magnetic disks, DASD, bubble memory; tape; optical disks such as CD-ROMs; and other persistent (also called nonvolatile) storage devices such as core, ROM, PROM, flash memory, or battery backed RAM. Those skilled in the art will appreciate that within the spirit and scope of the present invention, one or more of the components instantiated in the memory of the clients (60 . . . 63) or server (30 . . . 33) could be accessed and maintained directly via disk (260), the network 25, another server, or could be distributed across a plurality of servers.

Now that a preferred embodiment of the present invention has been described, with alternatives, various modifications and improvements will occur to those skill in the art. Thus, the detailed description should be understood as an example and not as a limitation. The proper scope of the invention is defined by the appended claims.

What is claimed is:

1. A method for identifying object fragments in an object, said method comprising the steps of:

analyzing an object description to identify one or more persistent object fragments associated with the object; creating the one or more persistent object fragments, in response to said analyzing; and

creating a persistent object fragment identity for a persistent object fragment, based on one or more of: formal descriptors; and an object fragment property.

2. The method of claim 1, wherein the object description is based on the formal descriptors, said method comprising the further steps of:

maintaining and tracking the persistent object fragment identity and associated formal descriptors; and generating a cacheable object fragment.

3. The method of claim 1, comprising the further steps of revising the object description by replacing at least one object fragment with an associated persistent object fragment identity to enable one or more of: the object fragment; and a revised object description to be cacheable at one or more of: a server; and a client; and sending a revised object description to the client.

4. The method of claim 3, wherein comprising the further steps of:

the client receiving and caching the revised object description; and

the client receiving a version of the one or more object fragments associated with the fragment identity, wherein the version is generated at the server and is based on the capability of the client.

11

5. The method of claim 1, further comprising the steps of:
receiving a request for an object fragment;
determining whether the fragment is cached, based on the
object fragment identity; and

if the fragment is not cached, dynamically generating the
fragment.

6. The method of claim 1, further comprising the step of
caching the object fragment based on one or more of: a
reference frequency; a cache size; and a processing cost.

7. The method of claim 1, further comprising the step of:
generating one or more different versions of the fragment;
wherein a version can be determined by one or more of:
a requesting device property and the fragment descrip-
tion.

8. The method of claim 7, further comprising the step of:
determining the version of the persistent fragment based on
the requesting device property and the fragment property.

9. The method of claim 1, wherein the fragment property
includes a processing requirement.

10. The method of claim 1, wherein the fragment property
includes one or more of a storage requirement and a band-
width requirement.

11. The method of claim 1, further comprising the steps
of:

identifying an object fragment as a dynamic object frag-
ment; and

transforming the dynamic object to a static object by
revising the object description and replacing one or
more dynamic object fragments with its object identity.

12. The method of claim 1, wherein the fragment property
includes whether the fragment can be generated efficiently
by various client devices.

13. The method of claim 1, wherein the formal descriptors
are markup tags in the object description and wherein the
object is described using a markup language.

14. The method of claim 1, wherein the object is a Web
page described using a markup language selected from the
group consisting of XML, SGML, or HTML.

15. The method of claim 1, wherein the object fragment
can be nested or hierarchical.

16. The method of claim 15, further comprising the steps
of:

recognizing a nested object fragment as including a
dynamic fragment or a frequently changed fragment;
and

making an outer fragment cacheable at one or more of a
server and a client.

17. The method of claim 1, further comprising the steps
of:

identifying one or more of the object fragments requiring
invalidation; and

garbage collecting invalid object fragments.

18. The method of claim 1, wherein the object fragment
property comprises the property selected from the group
consisting of: a dynamic property; a static property; how
frequently the object is going to change; size; or processing
cost to generate that fragment from its description.

19. The method of claim 1, further comprising the step of
caching the object based on one or more object fragment
properties.

20. The method of claim 1, further comprising the steps
of:

selecting a subset of the segments contained in the object;
and

recognizing the subset as persistent object fragments.

12

21. The method of claim 1, wherein said step of creating
a persistent object fragment further comprises the steps of:
recognizing and separating a segment as an object frag-
ment so it can be cached separately and reused to avoid
going through interpreting a markup language descrip-
tion of the object to generate it again; wherein the
segment will only be recognized as the object fragment
only if the segment or group of segments satisfies a
threshold for interpreting the markup language descrip-
tion based on one or more of: a processing requirement;
and a storage requirement.

22. The method of claim 1, wherein the persistent object
fragment will have a consistent identity regardless of
whether it appears in one or more of: multiple objects; and
multiple times in the same object.

23. A method for caching objects including object
fragments, said method comprising the steps of:

a client receiving from a server an object including a
revised object description wherein at least one object
fragment has been replaced with an associated persis-
tent object fragment identity based on one or more of:
formal descriptors; and an object fragment property, in
response to a request for the object; and

the client processing the revised object description.

24. The method of claim 23, further comprising the step
of:

the client receiving a version of the one or more object
fragments associated with the fragment identity,
wherein the version is generated at the server and is
based on the capability of the client.

25. The method of claim 24, wherein the version is
generated at the server and is based on the capability of the
client.

26. The method of claim 23, wherein the persistent object
fragment will have a consistent identity regardless of
whether it appears in one or more of: multiple objects; and
multiple times in the same object.

27. The method of claim 23, wherein the formal descrip-
tors are markup tags in the object description and wherein
the object is described using a markup language.

28. The method of claim 23, wherein the object is a Web
page described using a markup language selected from the
group consisting of XML, SGML, or HTML.

29. The method of claim 23, wherein said processing step
includes one or more of caching a revised object and
rendering the object.

30. The method of claim 23, further comprising the step
of: the client receiving from the server a version of the object
fragment interpret and generated at the server, wherein the
version generated is based on one or more of: the processing
power of the client; the storage capacity of the client; and the
bandwidth available to the client to retrieve a description of
the fragment.

31. The method of claim 23, wherein the persistent object
fragment identifier represents a dynamic object.

32. The method of claim 23, wherein the client is selected
from a group consisting of: a handheld device; a palmtop
device; a set-top box; a smart phone; or an Internet appli-
ance.

33. A program storage device readable by a machine,
tangibly embodying a program of instructions executable by
the machine to perform method steps for identifying object
fragments in an object, said method steps comprising:

analyzing an object description to identify one or more
persistent object fragments associated with the object;
creating the one or more persistent object fragments, in
response to said analyzing; and

13

creating a persistent object fragment identity for a persistent object fragment, based on one or more of: formal descriptors; and an object fragment property.

34. The program storage device of claim 33, wherein the object description is based on the formal descriptors, said method comprising the further steps of maintaining and tracking the persistent object fragment identity and associated formal descriptors; and generating a cacheable object fragment.

35. The program storage device of claim 33, comprising the further steps of

revising the object description by replacing at least one object fragment with an associated persistent object fragment identity to enable one or more of: the object fragment; and a revised object description to be cacheable at one or more of: a server; and a client; and sending a revised object description to the client.

36. The program storage device of claim 35, wherein comprising the further steps of:

the client receiving and caching the revised object description; and

the client receiving a version of the one or more object fragments associated with the fragment identity, wherein the version is generated at the server and is based on the capability of the client.

37. The program storage device of claim 33, further comprising the steps of:

receiving a request for an object fragment;

determining whether the fragment is cached, based on the object fragment identity; and

if the fragment is not cached, dynamically generating the fragment.

38. The program storage device of claim 33, further comprising the step of caching the object fragment based on one or more of: a reference frequency; a cache size; and a processing cost.

39. The program storage device of claim 33, further comprising the step of:

generating one or more different versions of the fragment; wherein a version can be determined by one or more of: a requesting device property and the fragment description.

40. The program storage device of claim 39, further comprising the step of: determining the version of the persistent fragment based on the requesting device property and the fragment property.

41. The program storage device of claim 33, wherein the fragment property includes a processing requirement.

42. The program storage device of claim 33, wherein the fragment property includes one or more of a storage requirement and a bandwidth requirement.

43. The program storage device of claim 33, further comprising the steps of:

identifying an object fragment as a dynamic object fragment; and

transforming the dynamic object to a static object by revising the object description and replacing one or more dynamic object fragments with its object identity.

44. The program storage device of claim 33, wherein the fragment property includes whether the fragment can be generated efficiently by various client devices.

45. The program storage device of claim 33, wherein the formal descriptors are markup tags in the object description and wherein the object is described using a markup language.

14

46. The program storage device of claim 33, wherein the object is a Web page described using a markup language selected from the group consisting of XML, SGML, or HTML.

47. The program storage device of claim 33, wherein the object fragment can be nested or hierarchical.

48. The program storage device of claim 15, further comprising the steps of:

recognizing a nested object fragment as including a dynamic fragment or a frequently changed fragment; and

making an outer fragment cacheable at one or more of a server and a client.

49. The program storage device of claim 33, further comprising the steps of:

identifying one or more of the object fragments requiring invalidation; and

garbage collecting invalid object fragments.

50. The program storage device of claim 33, wherein the object fragment property comprises the property selected from the group consisting of: a dynamic property; a static property; how frequently the object is going to change; size; or processing cost to generate that fragment from its description.

51. The program storage device of claim 33, further comprising the step of caching the object based on one or more object fragment properties.

52. The program storage device of claim 33, further comprising the steps of:

selecting a subset of the segments contained in the object; and

recognizing the subset as persistent object fragments.

53. The program storage device of claim 33, wherein said step of creating a persistent object fragment further comprises the steps of:

recognizing and separating a segment as an object fragment so it can be cached separately and reused to avoid going through interpreting a markup language description of the object to generate it again; wherein the segment will only be recognized as the object fragment only if the segment or group of segments satisfies a threshold for interpreting the markup language description based on one or more of: a processing requirement; and a storage requirement.

54. The program storage device of claim 33, wherein the persistent object fragment will have a consistent identity regardless of whether it appears in one or more of: multiple objects; and multiple times in the same object.

55. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for processing objects including object fragments, said method steps comprising:

a client receiving from a server an object including a revised object description wherein at least one object fragment has been replaced with an associated persistent object fragment identity based on one or more of: formal descriptors; and an object fragment property, in response to a request for the object; and

the client processing the revised object description.

56. The program storage device of claim 55, further comprising the step of:

the client receiving a version of the one or more object fragments associated with the fragment identity, wherein the version is generated at the server and is based on the capability of the client.

15

57. The program storage device of claim 56, wherein the version is generated at the server and is based on the capability of the client.

58. The program storage device of claim 55, wherein the persistent object fragment will have a consistent identity 5 regardless of whether it appears in one or more of: multiple objects; and multiple times in the same object.

59. The program storage device of claim 55, wherein the formal descriptors are markup tags in the object description and wherein the object is described using a markup lan- 10 guage.

60. The program storage device of claim 55, wherein the object is a Web page described using a markup language selected from the group consisting of XML, SGML, or 15 HTML.

61. The program storage device of claim 55, wherein said processing step includes one or more of caching a revised object and rendering the object.

16

62. The program storage device of claim 55, further comprising the step of: the client receiving from the server a version of the object fragment interpret and generated at the server, wherein the version generated is based on one or more of: the processing power of the client; the storage capacity of the client; and the bandwidth available to the client to retrieve a description of the fragment.

63. The program storage device of claim 55, wherein the persistent object fragment identifier represents a dynamic object.

64. The program storage device of claim 55, wherein the client is selected from a group consisting of: a handheld device; a palmtop device; a set-top box; a smart phone; or an Internet appliance.

* * * * *